Study of Evolutionary and Swarm Intelligent Techniques for Soccer Robot Path Planning

Mostafa E. Salehi *, Elahe Mansury

Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

Received 15 April 2016; revised 28 October 2016; accepted 16 January 2017; available online 16 March 2017

Abstract

Finding an optimal path for a robot in a soccer field involves different parameters such as the positions of the robot, positions of the obstacles, etc. Due to simplicity and smoothness of Ferguson Spline, it has been employed for path planning between arbitrary points on the field in many research teams. In order to optimize the parameters of Ferguson Spline some evolutionary or intelligent algorithms are proposed in the literature. In this paper, we present a comparative study on different evolutionary and swarm algorithms as solutions to the problem of robot path planning. We optimize the parameters of Ferguson Spline and find the best path between two arbitrary points, studying Differential Evaluation (DE), Genetic Algorithm (GA), Evolutionary Strategies (ES), Artificial Bee Colony (ABC), and Particle Swarm optimization (PSO) algorithms. Firstly, a path for robot movement is describe by Ferguson splines and then these algorithms are used to optimize the parameters of splines to find an optimal path between the starting and the goal point considering the obstacles between them. The experimental results show the performance and effectiveness of the studied solutions in comparison with other swarm intelligent algorithms.

Keywords: *Path planning, Ferguson Splines, Humanoid soccer playing robot, Swarm Intelligent.*

1. Introduction

Development of robots and how to make them more intelligent have been a significant scientific issue for researchers in this field. Researches concerning robotic matters have always been challenging and it is getting more complicated day by day [1]. During several years ago, designing robots capable of playing soccer with human has been one of the promising goals of researchers in this field. Many researches have been carried out to improve the ability of robots to play physically and intelligently as much alike human as possible [2]. Among those efforts, moving a robot over an optimal path through the field while avoiding other robots as obstacles and getting to the ultimate goal position in an acceptable time was the main concern.

In this paper, we propose a solution to the problem of path planning using Differential Evaluation (DE) algorithm, Genetic Algorithm (GA), Evolutionary Strategies (ES), and cubic Ferguson Splines. In Section 2, we de-scribe the Ferguson splines, a fitness function that is used for optimization, and how parents are coded for finding the optimum solution. Experimental results are presented in section 3 to reflect the performance of the algorithms in finding optimal path. Finally, paper is concluded in section 4.

* Corresponding author. Email: mostafa.salehi@gmail.com

2. Robot Path Planning and Ferguson Splines

2.1. Ferguson Splines

Ferguson spline was first introduced by J.C. Ferguson in 1964. It is a special form of piece-wise cubic spline and formed by two operative (starting and end) points as well as two vectors based on these points [3]. Considering P_0 and P_1 as the beginning and end points of the spline and P_0' and P_1' as their corresponding tangent vector, Ferguson spline can be established using the following equation:

$$X(t) = P_0 f_1(t) + P_1 f_2(t) + P_0' f_3(t) + P_1' f_4(t) \quad (1)$$

Where $t \in [0,1]$ and corresponding basis function f_1, f_2, f_3, f_4 are Fergusonmulti-nomials, which are given by:

$$f_1(t) = 2t^3 - 3t^2 + 1 \quad (2)$$

$$f_2(t) = 2t^3 + 3t^2 \quad (3)$$

$$f_3(t) = t(t-1)^2 \quad (4)$$

$$f_4(t) = t^2(t-1) \quad (5)$$

P_0 And P_1 can be obtained by $X(0)$ and $X(1)$. Values of points P_0' and P_1'aresimply obtained by substitution of derivations

$$f_1'(t) = 6t^2 - 6t \quad (6)$$

$$f_2'(t) = -6t^2 + 6t \quad (7)$$

$$f_3'(t) = 3t^2 - 4t + 1 \quad (8)$$

$$f_4'(t) = 3t^2 - 2t \quad (9)$$

Using equations (1) and (6 – 9), we can conclude that $P_0' = X'(0)$ and $P_1' = X'(1)$.

In case of another Ferguson spline:

$$\overline{X}(t) = \overline{P}_0 f_1(t) + \overline{P}_1 f_2(t) + \overline{P}_0' f_3(t) + \overline{P}_1' f_4(t) \quad (10)$$

We can describe a path using a string of Ferguson splines by connecting these two splines employing equation (11). The continuity of the first derivation is mandatory.

$$P_1 = \overline{P}_0', P_1' = \overline{P}_0' \quad (11)$$

2.2. B. Coding and Fitness Function

To simplify the problem of path planning, mathematic notation of Ferguson spline in 2D space can be described as:

$$r(t) = (x(t), y(t)) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \quad (12)$$

Where

$$\begin{cases} a_0 = 2P_0 - 2P_1 + P_0' + P_1' \\ a_1 = -3P_0 + 3P_1 - P_0' + P' \\ a_2 = P_0' \\ a_3 = P_0 \end{cases} \quad (13)$$

As mentioned previously, in cubic Ferguson spline, each spline is described by two points (beginning and end) P_0 and P_1 as well as two tangent vectors P_0' *and* P_1'. Regarding equation (11), each pair of neighboring splines within a string, shares one of the terminal points and the corresponding vector. Therefore, to define the whole trajectory in a 2D space, $4(n+1)$ variables are needed, where n is the number of spline in the string. According to this representation, the structure of parameters within the optimizing problem is illustrated in Figure 1.

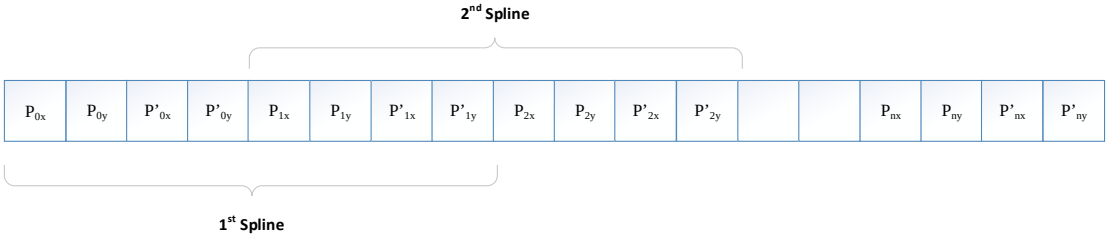

Fig. 1.Structure of parameters in the optimization problem.

To find the best solution for an optimization problem, choosing a good evaluation function plays an important role. Several different fitness functions have been proposed in the literature for the problem of path planning. However, in this paper, we use the one proposed in [3], which has two characteristics. First, it penalizes the long distance trajectories; in other words, the fitness function is inversely proportional to the length of the trajectories as defined in equation (14):

$$f_1 = \frac{l}{l_{min}} \tag{14}$$

Where l_{min} is a constant and equals to Euclidian distance between actual and desired robot position and l is the length of trajectory.

Second, the proposed function also penalizes those trajectories that lead to collisions with obstacles. By defining a safe distance, fitness function tries to consider trajectories that have minimal distance d_{min}which are greater than threshold value. To provide such option, the function is sharply inversely proportional to the minimal distance as defined in equation (15):

$$f_2 = \begin{cases} L, & d_{min} > d_{safe} \\ e^{\frac{D_{safe+1}}{d_{min}+1}+1} & d_{min} \leq d_{safe} \end{cases} \tag{15}$$

By considering both situations for having a good fitness function, the final function is defined as:

$$f = f_1 + \alpha f_2 \tag{16}$$

Where α is a weight factor that adjusts the proportion of the length and safety?

3. Optimization Algorithm

3.1. Evolutionary Strategies (ES)

Evolutionary strategy is among the first evolutionary algorithms. Based on numerical optimization, it creates n-dimensional vector with real values representing possible solutions. Mutation is applied considering equal standard deviation and the candidates will be evaluated and selected for the next round. In simplest form of selection, (1+1)-selection, parents will recombine to form a new offspring which eventually replaces the worst parent. The operation for ES is summarized below [4]:

1: Initialize Population
2: ***repeat***
3: Recombination
4: Mutation
5: Evaluation
6: Selection
7: ***until*** *requirements are met*

3.2. Differential Evaluation (DE)

Differential Evolution is another population-based algorithm. It employs the similar operators as GA including crossover, mutation and selection by emphasizing on mutation instead of crossover to create better result. The algorithm starts with a set of random population representing candidate solution to the problem. It utilizes the mutation operator for finding the best solution based on the differences of randomly selected solution pairs. It also uses

crossover operators to search for better solution. Selection leads the search toward the area with potential candidates [16]. The phases of the algorithm are as follows:

1: Initialize Population
2: Evaluation
3: ***repeat***
4: Mutation
5: Recombination
6: Evaluation
7: Selection
8: ***until*** *requirements are met*

3.3. Particle Swarm Optimization (PSO) and Clerc's PSO (CPSO)

Particle Swarm Optimization was introduced by Kennedy and Eberhart [16]. It is a population-based search algorithm and is designed by inspiring from social behavior of birds within a flock. In PSO, individuals, which are called particles and represent the solutions in the search space are flown and change their positions inside the space on a tendency to improve success of individuals and the swarm. The position of each particle is influenced by its own experience and of its neighbors. The general algorithm for PSO and CPSO is as follows:

1: Initialize Population
2: ***repeat***
3: Calculate fitness values of particles
4: Modify the best particles in the swarm
5: Choose the best particle
6: Calculate the velocities of particles
7: Update the particle positions
8: ***until*** *requirements are me*

The difference between PSO and CPSO is that a coefficient is used to provide the balance between exploration and exploitation behavior of algorithm in CPSO.

3.4. Artificial Bee Colony (ABC)

The Artificial Bee Colony is a swarm intelligence technique that was proposed by Karaboga in 2005. The driving objective behind the algorithm comes from intelligent foraging behavior of honey bees in which bees search out for food sources in nature in the best possible way. The ABC model consists of three different types of bees co-existing in a colony and cooperating together to search, find, and provide food: Employed bees, On-lookers, and Scouts. While employed bees bring foods from different sources into the hive and communicate the situations with other bees, an onlooker chooses a food source on the dancing information of employed bees. Finally, scouts try to find new sources in a random manner. The model starts with a population of individuals (foods), representing potential solutions. Then scout and onlooker bees try to direct the search toward finding the best solution by exploiting existing food sources and looking for other sources in the surrounding environment. Scout bees will shift the searching direction to other areas in the solution space once in a while. The position of a food source in the problem space indicates a solution, and the nectar amount of it represents the quality (fitness) of the solution. The ABC algorithm is presented as follows [17]:

1: Initialization Phase
2: ***Repeat***
3: Employed Bees Phase
4: Onlooker Bees Phase
5: Scout Bees Phase
6: Memorize the best solution achieved so far
7: ***Until*** *(Cycle=Maximum Cycle Number)*

3.5. Genetic Algorithm (GA)

Genetic algorithm is one of the fundamental algorithms of evolutionary computing. Like many methods of this category, it contains essential components for finding the best solution such as

fitness evaluation as well as genetic operators such as reproduction, crossover, and mutation. GA starts with an initial population of candidate solutions in form of number strings (usually binary strings). A fitness value is computed by fitness function and assigned to each string. The algorithm continues with evaluating the population and through reproduction operation, which tries to select the best possible candidates for the next round; strings with higher fitness values have more chance to be selected. By applying the crossover operation, new members will be added to the population. By choosing a pair of strings randomly and create new pairs, crossover move the algorithm forward to find the best solution. Meanwhile, mutation operator, mutates the string to either change the pace when the algorithm falls into local minima or to explore new area for finding the best solution [16]. The main phases of the algorithm are as follows:

1: Initialize Population
2: ***repeat***
3: Evaluation
4: Reproduction
5: Crossover
6: Mutation

4. Experimental Results and Discussion

4.1. Settings

In the experiments, for fair comparisons, values of common parameters used in each algorithm such as population size and number of iterations were the same. For each algorithm some specific settings were needed to be considered, which are given in Table 1. In our experiment, Population = 20, Cycle = 100, d_{safe}= 0.6, number of spline (n) = 2, upper-bound = 3 and lower-bound = -1.Reported results are based on 10 different runs of each algorithm.

Specific Settings: Values of specific parameters for each algorithm are given in Table 1. Best Parameters for each algorithm are chosen after performing several runs and comparing the results for that algorithm.

Table. 3. Specific parameters of each algorithm

Algorithms	Parameters	Value	Parameters	Value
CPSO	*Min and Max Value*	*0, 5*	*c1 and c2*	*1*
	inertia weight	*1*	*Colony Size*	*20*
DE	*Min and Max Value*	*0, 2*	*Weighting factor*	*0.7*
	Crossover constant	*0.3*		
ES	*Min and Max Value*	*-1, 3*	*Lambda*	*10*
	phiCount	*0*	*Sigma*	*1*
	Colony Size	*20*		
GA	*Min and Max Value*	*0, 0.5*	*Crossover type*	*Single point*
	Crossover probability	*1*	*Mutation probability*	*0.01*
	Colony Size	*20*		
ABC	*Min and Max Value*	*3, -3*	*Limit*	*30*
	Colony Size	*20*		

4.2. Experiments

The robot must travel a path from an arbitrary point inside its own half-field toward another arbitrary point close to the opponent goal (Figure 2).

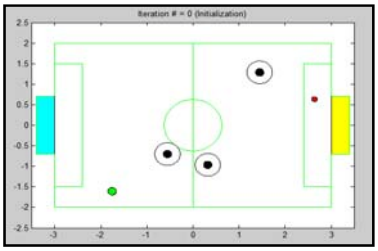

Fig. 2. Models to find an optimal path

All algorithms converged around 2 and 2.01 meters after several iterations. However, CPSO, BBO, ES, arrived to these points between iterations 2 and 4 which is faster than DE (between iterations 6 and 12).

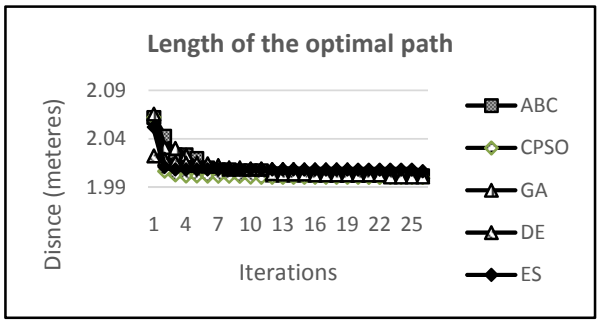

Fig. 3.Optimization of the path by 6algorithms for first 30 iterations

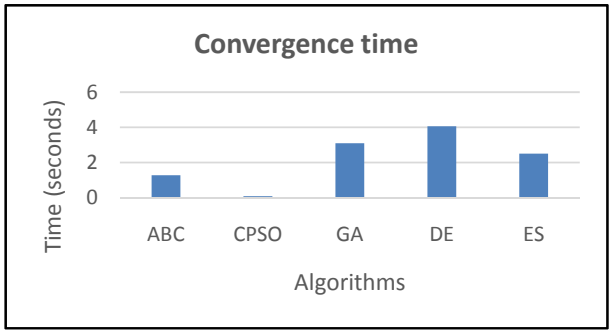

Fig. 4.Time of each algorithm for the experiment

According to the figure, CPSO outperforms all the other algorithms in finding the optimal path in the mini-mum number of iterations and shortest time. PSO and ES, behave similarly in case of convergence time. Finding the optimal path in a fewer number of iterations with longer running time or vice versa. DE performed poorly .it has the convergence time nearly twice the others. ABC, GA performances were better.

5. Conclusion

In this paper, we study the path planning algorithms for finding an optimal path for a soccer robot traveling between two arbitrary points inside the game field considering no collisions with obstacles. We employed Ferguson Cubic Spline that its parameters are optimized with the proposed algorithms.

After executing different scenarios and examining the behavior of different algorithms, we decided to com-pare them based upon convergence time. CPSO and ES were the fastest ones on average. CPSO outperforms other algorithms in considerable amount of time and DE is the slowest one. DE performance was half of the others. Therefore, for the robot path planning problem, considering both convergence iterations and time, we suggest CPSO among Differential Evaluation, Genetic Algorithm, Evolutionary Strategies, Artificial Bee Colony (ABC) algorithms.

References

[1] Lee, T. K.; Baek, S. H.; Choi, Y. H.; Oha, S. Y., “Smooth coverage path planning and control of mobile robots based on high-resolution grid map representation, ”International Journal of Robotics and Autonomous Systems, ELSEVIER, pp: 801-812, (2011).

[2] Raja, P.; Pugazhenthi,S., ”Optimal path planning of mobile robots: A review”, International Journal of Physical Sciences Vol. 7(9), pp: 1314 - 1320 (2012).

[3] Saska, M.; Macas, M; Preucil, L., “Robot path planning using particle swarm optimization of Ferguson splines,” Proceedings of the IEEE Symposium on Emerging Technologies and Factory Automation, pp: 833-839 (2006).

[4] Raja, P.; Pugazhenthi,S., "Path planning for a mobile robot in a dynamic environment " International Journal of the Physical Sciences, pp: 4721-4731 (2011).

[5] Khatib, O., "Real-time obstacle avoidance for manipulators and mobile robots," Proceedings. IEEE International Conference in Robotics and Automation. pp: 500-505, (1985).

[6] Mansury, E.; Nikookar, A.; E.Salehi, M., “Artificial Bee Colony optimization of ferguson splines for soccer robot path planning”, First RSI/ISM International Conference on Robotics and Mechatronics (ICRoM), pp: 85-89 (2013).

[7] Mansury, E.; Nikookar, A.; E.Salehi, M., “Differential Evolution Optimization of Ferguson Splines for Soccer Robot Path Planning”, Symposium on Artificial Intelligence and Signal Processing (AISP), pp: 311-319 (2013).

[8] Karaboga ; Basturk, B., “A comparative study of Artificial Bee Colony algorithm”, Journal of Applied Mathematics and Computation, pp: 108–132 (2009).